



# Common Threads in Life

A human being should be able to change a diaper, plan an invasion, butcher a hog, conn a ship, design a building, write a sonnet, balance accounts, build a wall, set a bone, comfort the dying, take orders, give orders, cooperate, act alone, solve equations, analyze a new problem, pitch manure, program a computer, cook a tasty meal, fight efficiently, die gallantly. Specialization is for insects.

—Robert A. Heinlein, *Time Enough For Love* [1]

Looking for patterns, themes, and repeated motifs is a common technique for understanding many subjects. Gamma [2] did a great job describing the most proven software development patterns. The Software Program Managers Network (SPMN) [3] has captured dozens of lessons learned in managing projects. But like the late Robert Heinlein, my focus is a bit broader than those examples.

Fifteen themes and patterns have emerged (so far!) from wildly disparate activities. Fields as diverse as dancing, project management, golf, engineering, and massage therapy have contributed to this collection of observations about how we think, plan, move, and analyze. Here are 11 of 15 observations most relevant to software engineering and project management.

1. **Focus beyond what is possible or seems immediately relevant.** Golf, tennis: Continue the swing smoothly long past contact with the ball.  
Karate: Focus the target of a technique farther than you will actually be able to strike.  
Dance: Focus your attention and maintain connection beyond your body – into the earth, into the sky, with your partner, to the audience.
2. **Focus attention on the desired outcome, not what you're avoiding.**  
Golf: The best way to hit the ball poorly is to focus on what you don't want the ball to do.  
Project management: A project plan identifies the tasks needed to achieve the desired objective of the project.  
Karate: Assume you will be successful and determine how to make it happen.
3. **Recognize risks without dwelling on them.**  
Project management: sound risk management is critical to success, but can't be the only activity.  
Golf: plan for likely errors, without falling into the trap of the previous observation.  
Karate: Recognize your opponent's strengths, yet plan your strategy for success in spite of them.
4. **When something goes well, stick to the basics.**  
Project estimation, karate, golf, piano, dance: Practicing the basics is the key to achieving better performance.
5. **When something goes badly, go back to the basics.**  
Project management: A classic mistake is to abandon the project plan when something goes wrong – instead, that's the time to return to basic understanding of tasks to be done and measuring progress toward achieving them [4].  
Golf: After a bad shot, the best way to avoid a string of more bad shots is to focus on basic technique.
6. **Follow by rote at first; then with experience, tailor your approach.**

Project management, dance, karate, massage, construction, etc: When first learning a new skill, it is common to follow literally a prescribed set of actions. As you develop more skills, you develop the ability to adopt and blend techniques from a variety of sources.

7. **Attention to detail separates good from great.**  
Carpentry, fashion, music, dance: The difference between ordinary work and excellent work is often in attention to details.
8. **Balance similarity and opposites.**  
Project management, engineering: Most management and design decisions involve balancing conflicting needs (speed vs. quality, light vs. strong, etc.), yet good design practice encourages reuse and application of patterns.  
Dance, music: Use changes between fast and slow, smooth and sudden, harmony and dissonance, symmetry and asymmetry, and repetition and novelty to create interesting work.
9. **Left field is a good place to visit often.**  
Program management: Successful contract approaches often use unconventional structure.  
Engineering, science: Many great insights have come from pulling together seemingly unrelated concepts and discovering synergy among them.
10. **All things are rarely equal.**  
Risk management: Risks need to be quantified to see which are most important.  
Software engineering: need to tailor the scope of testing to match the complexity and criticality of the component.  
System modeling: need to model the critical aspects of a system and set aside the rest.
11. **Examine a problem with different sets of eyes.**  
Requirements engineering: Get input from all kinds of users and stakeholders.  
Astronomy: Insights have been obtained from looking the same direction, but using visible, infrared, or polarized light, or using a radio telescope.  
Bioengineering: Learn about a material by correlating different aspects of it – physical properties, chemical reactions, luminescence, and radioactivity.  
The other four observations relate mostly to physical movement and are in the online version of this document <[www.stsc.hill.af.mil/crosstalk/2007/08/0708/backtalk.html](http://www.stsc.hill.af.mil/crosstalk/2007/08/0708/backtalk.html)>. This a starting point for continued thought and evaluation – feedback, rebuttal, and additions are welcome!

— Glenn Booker  
gbooker@drexel.edu

## References

1. Heinlein, Robert A. *Time Enough for Love: The Lives of Lazarus Long*. New York: Ace, 1994.
2. Gamma, Erich, Richard Helm, Ralph Johnson, and John Vlissides. *Design Patterns, Elements of Reusable Object-Oriented Software*. Indianapolis: Pearson, 1995.
3. SPMN Guidebooks. *Integrated Computer Engineering*. 2006 <[www.spmn.com/pdf\\_download.asp](http://www.spmn.com/pdf_download.asp)>.
4. McConnell, Steve. "Rapid Development." Redmond, WA: Microsoft Press, 1996.